enior Design

ENG EC 463

Final Test Report

To: Professor Pisano

From: Team 17

Team: Future of Heat

Team Members: Fatima Dantsoho, Andrew Chen, Ben Gross, Muayad Al Riyami, Seunghak Shin

Date: 4/9/21

Subject: Final Testing Report

1.0 Required Materials

- **Hardware**
  - Function generator
  - DC power supply
  - Oscilloscope
  - Cables/wires
  - Town load units (PCB boards)
  - Arduino Mega 2560 R3
  - ESP8266 Feather Huzzah
  - Breadboarded Demultiplexer Circuits
    - 8x CD4053BMT DEMUX
    - 24x 330 Ω Resistors
  - 2x 4AA Battery packs

- **Software**
  - **HTML 1 script**
    - Front end page
  - **JavaScript 2 scripts**
    - Create output graphs
    - Read from and Write to firebase database
  - **Firebase Database**

2.0 Set Up

- **Web Application:** The web application is the main user interface. The user makes a selection of the adoption rate of electric homes and eclectic vehicles and the weather condition. The user input is used to calculate the amount of houses and vehicles on the diorama that will be heated electrically and the calculated values will be pushed to the firebase database. The arduino updates the load profiles of each house based on the user inputs and the updated values are read by the web application and load, cost and CO_2 Emission of each house is plotted in the output section.

- **Circuit:** Connect the transformer to the circuit and plug it into a wall socket. Using the oscilloscope, probe the voltage at the node between the load resistor and the AC relay. Now, all that is left is to apply the DC voltage to turn a desired relay on. We will test if the house unit PCB we built works as intended by demonstrating that each load branch is controlled by the relays.

- **Arduino Mega + ESP8266 Component:** Turn on all the battery packs attached to the front of the diorama, by flipping the power switches to ON, refer to **Figure.7**. Connect the USB power cables for the Arduino Mega and the ESP8266 to an accompanying laptop. To set up the Wifi connection for the ESP8266 edits to the Wifi SSID and Wifi password sections of the ESP8266 Github code should be done, reupload this code to the ESP8266 through the Arduino IDE. The simplest method of connecting to Wifi will be setting up a mobile hotspot on the current Wifi network, remember to set the network band to 2.4 GHz. Upon ensuring the Arduino Mega and ESP8266 are properly powered and connected, this project portion is ready for testing.

3.0 Pre-Testing Setup Procedure

- **Web Application**
  1. Initialize firebase database using terminal - $firebase init
  2. Open the index.html file with the web browser (Google Chrome).

- **Arduino Mega + ESP8266 Component:**
  1. Dim the lights to visualize the LEDs clearer.

4.0 Testing Procedure

- **Web Application**
    1. Open the index.html file with the web browser (Google Chrome).
    2. Press the “Click to Start” button to begin.
    3. Select input for electric home and vehicle adoption rate and weather using the range slider
    4. Click the Enter button to process and upload inputs to the firebase and receive the load profiles for each house.
    5. The Load profile array and CO_2 Emission, Cost and Load profile Line graphs should be displayed in the “Output” section.

- **Circuit**
    1. The success of the test relies on whether the house unit PCBs are built correctly according to design and that the design is functional. The main goal of the house unit circuit is to control the load (allow current to flow through or not) using voltage controlled AC relays.
    2. Apply 5 volts to a specific relay and probe the voltage of the load resistors using the oscilloscope.
    3. If the relay is off (open circuit), then the oscilloscope reading should be the same as the function generator output (no voltage drop across load resistor). If the relay is on (closed circuit), then the oscilloscope reading should be approximately zero as it is now connected to ground.

- **Arduino Mega + ESP8266 Component**
    1. Ensure that the ESP8266 module is working as intended and that there is communication between the Arduino Mega and the Web app.
    2. As the user changes the parameters for the percentage of electric heating adoption, percentage of electric vehicle adoption, and temperature severity on the Web App, the LEDs mounted on and within the models houses should correspondingly change.

5.0 Measurable Criteria

- **Web Application**
    1. Database resets all values to 0 when the “Click to start” button is clicked.
    2. Databases should be automatically updated once the inputs are entered.
    3. Load Profile output should be displayed as an array in the output section.
    4. All graphs should be populated.
- **Circuit**

1. Current passing through the electric heating branch.
2. Current passing through the electric vehicle branch.
3. A successful test should have currents passing through the loads when the relay control voltage is high and none through them when it is low.

- **Arduino Mega + ESP8266 Component**
  1. R/G LEDs mounted on the house roofs should correspond to Web App input of ⌊% Electric Heating adoption * Total Number of Heating Units⌋. This is since the total number of heating units is 17, rounding down should decimals appear.
  2. R/B LEDs mounted within the houses should simulate houses heating up. This heating up rate is dependent on the inputted weather severity and fixed house size.
  3. Note that currently there are 11 residential homes and 1 hotel. One home signifies a heating load and an electric vehicle load, while the hotel signifies six heating loads and six electric vehicles. When adoption exceeds 50% the hotel load is placed in. E.g. Electric heating for hotel (6 units) and two other houses (2 units) for a total of 8 units is ⌊50% * 17 units⌋, more specifics are written in **Figure.7**.

6.0 <u>Score Sheet</u>

Web Application

User Input			**Result**			**Correct? (Y/N)**
electric homes adoption rate (%)	**electric vehicles adoption rate (%)**	**Weather (F)**	**# of homes**	**# of EV**	**Weather Severity**	
50	10	10	**6**	**1**	**5**	**Y**
40	50	20	**4**	**8**	**4**	**Y**
70	30	30	**8**	**5**	**3**	**Y**
20	40	40	**2**	**6**	**2**	**Y**
80	70	50	**9**	**11**	**1**	**Y**

Result All tests were passed and updated automatically to the web database	

Circuit

	Current through Electric Heating	Current through Electric Vehicle
Relay control voltage - High	Yes	Yes
Relay control voltage - Low	No	No

Arduino Mega + ESP8266 Component

Red/Green Dual LED Metrics		**Red/Blue Gradient (Heating Visual)**	**Time Delay <2 sec LED change? (Y/N)**
# Electric Homes Input ⌊%*17⌋	**Correct Corresponding LEDs? (Y/N)**	**Visual differences in "heating times" depending on weather severity input and on module sizes?**	
5	Yes	Visual differences in between the time it takes to heat up each house based on house size and temperature is apparent but only when one intently looks for it. Mainly since the LEDs don't emit diffused light.	This current test uses the Arduino serial input as a substitute for the ESP8266, but yes, time delay < 2 sec.
8	Yes		
15	Yes		
17	Yes		
Result: Integration of LEDs onto the diorama was a success. LED control works but without the ESP8266 component.			

7.0 <u>Conclusion</u>

- **Web Application**

The web application was able to successfully calculate the amount of electric houses and electric vehicles using the adoption rates the user picked and also the appropriate weather severity range. The database was immediately updated after the enter button was pressed and random values were generated to represent each house load profile. The load profile for each house was printed on the web application starting from Load 1 to Load 12.

- **Circuit**

The circuit testing was successful because the AC relays were able to control the current through the electric heating and vehicle branches. The voltages at the oscilloscope nodes labeled in **Figure.1** were high when the relay input voltage was at a low, indicating that there is no voltage drop across the resistor. Therefore, no current is flowing through the branch. When the relay input voltage was high, the voltages at the oscilloscope nodes were low for the branches indicating that current is flowing since there is a voltage drop across the resistor. **Figure.5** shows the oscilloscope reading when the relay input voltage is low and **Figure.6** shows the reading when the voltage is high.

- **Arduino Mega + ESP8266 Component**

The current Arduino component as mentioned above is incomplete. Issues in timing the serial communication alongside the LED control resulted in errors as such the wifi portion was omitted. Regardless, as seen in the testing score sheet above the current deliverables demonstrate that the LED control for the entire diorama is complete. The R/G and R/B LEDs do correctly reflect changes to the houses given serial console inputs for (1) number of electric heated homes and (2) weather severity. Additionally, the input latency isn't an issue although currently this is inaccurate as the wifi communication hasn't come into the picture. Even so parsing the serial console for the pseudo inputs also takes quite a bit of time.

8.0 Hardware Pinout

Arduino Mega Pinout (Complete: All Modules)	
Pin #	Usage Description
	Control Pins for R/G LEDs -> R/G DeMux Pins (9-11)

D27, D28, D29 D30, D31, D32 D38, D39, D40 D41, D42, D43	Control Pins for Module 1 Control Pins for Module 2 Control Pins for Module 3 Control Pins for Module 4
 D9 D7 D5 D3	Control Bit for R/B LEDs -> R/B DeMux Pins (9-11) Control Pins for Module 1 Control Pins for Module 2 Control Pins for Module 3 Control Pins for Module 4
 D8 D6 D4 D2	PWM Pin for R/B LEDs -> R/B DeMux Pins (4,14,15) Control Pins for Module 1 Control Pins for Module 2 Control Pins for Module 3 Control Pins for Module 4
D13	Enable signal for DeMux -> Pin 6 on all DeMux
A0, A1	Negative and positive terminal probing RMS voltage respectively
RX1,TX1	Serial communication with ESP8266 Module
CD403BMT Demultiplexer: DeMux (Complete)	
1,2,3,5,12,13	Output pins to 330 Ω resistors then to R/G and R/B LED terminals
4,14,15	Input voltage from 6V battery pack/ PWM Arduino signal depending on if it's for R/G or R/B.
6	Enable pin for DeMux
7,8	Ground
9,10,11	Control bits for demultiplexing
16	Power input for DeMux -> 6V battery pack
ESP8266: Wifi Module	
RX,TX	Serial Communication with Arduino Mega

<u>Table 1 .1</u>

This pinout details all of the pin connections between the Arduino Mega, Demultiplexers, and the ESP8266. Note that the 6V AA Battery pack used to power the CD403BMT Demultiplexer will also be used to power the R/G LED modules, with 1 battery pack powering 2 sub modules (the LEDs and the demux) respectively. Additionally, it isn't apparent here but the Arduino Mega and the ESP8266 modules are powered via USB cables that will be plugged into the diorama's accompanying laptop.

*Sidenote: As all of the circuitry and wires for the LEDs need to be passed through the foamcore platform all the houses are mounted on, breadboards were used to wire up the necessary circuits. This can be seen in **Figure.8.**

9.0 Appendix

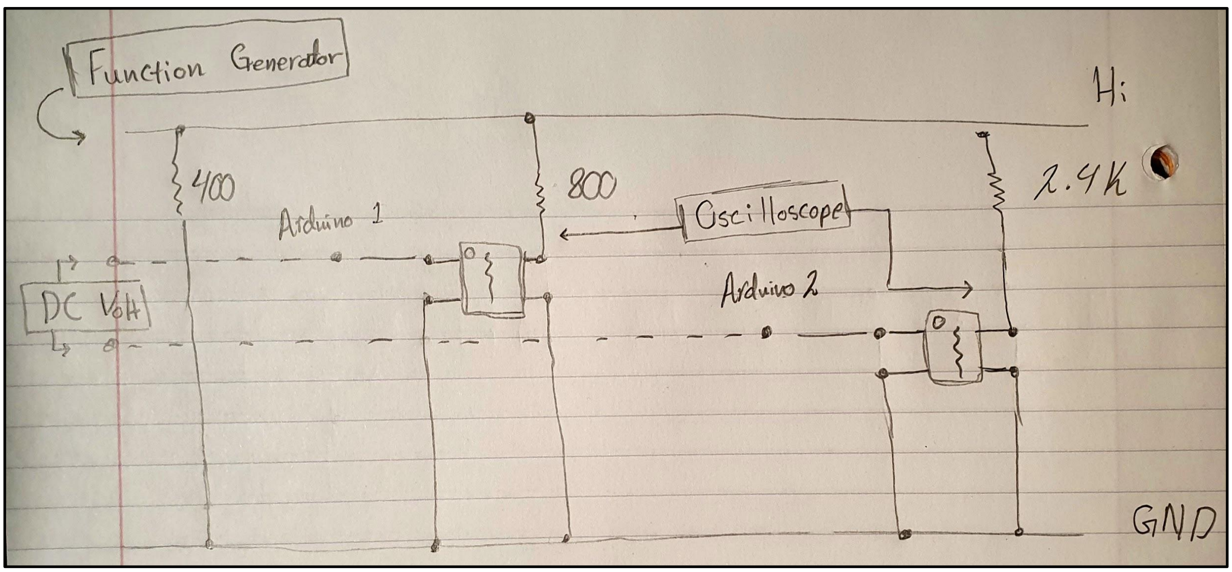


Figure.1 : House Load Testing Schematic

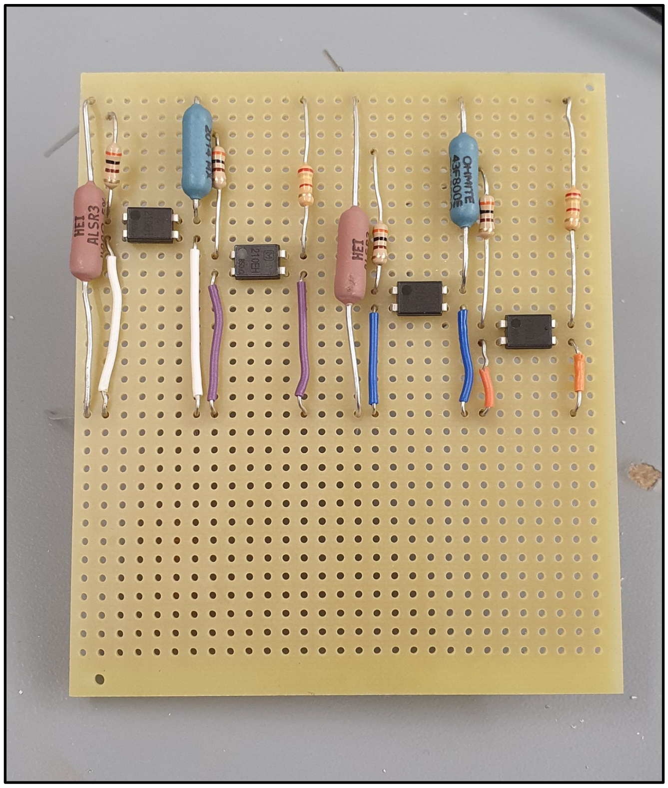


Figure.2 : House Load PCB (two units on one board)

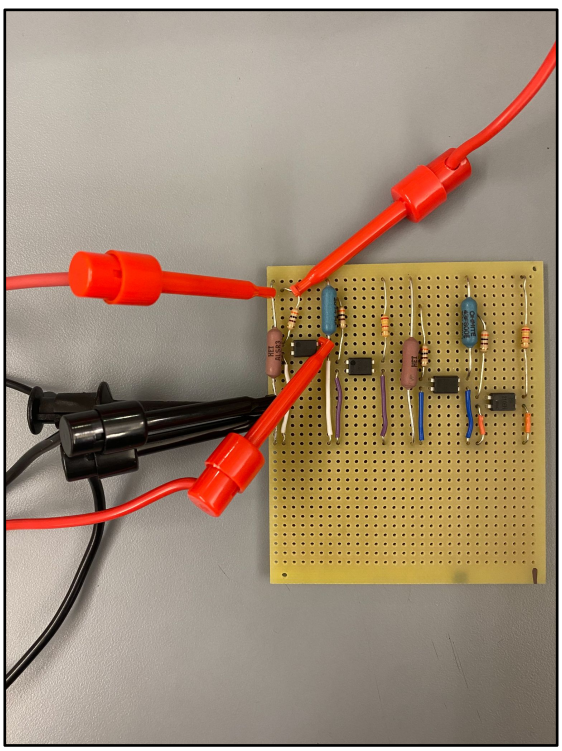


Figure.3 : Electric Heating Testing Configuration

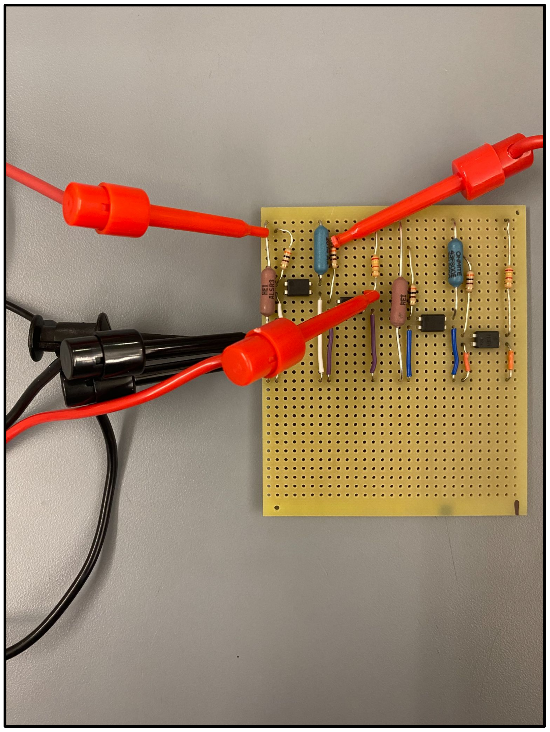


Figure.4 : Electric Vehicle Testing Configuration

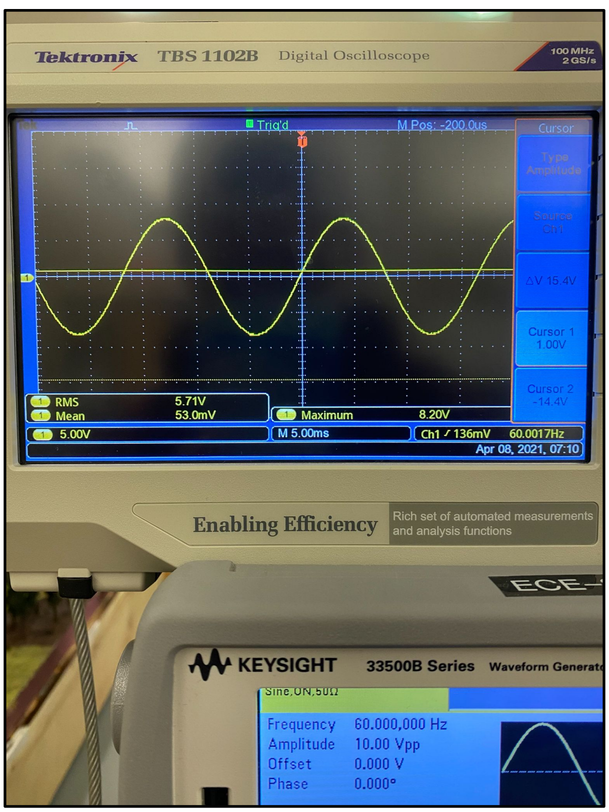


Figure.5 : Voltage at Oscilloscope Node (Relay Control Voltage - Low)

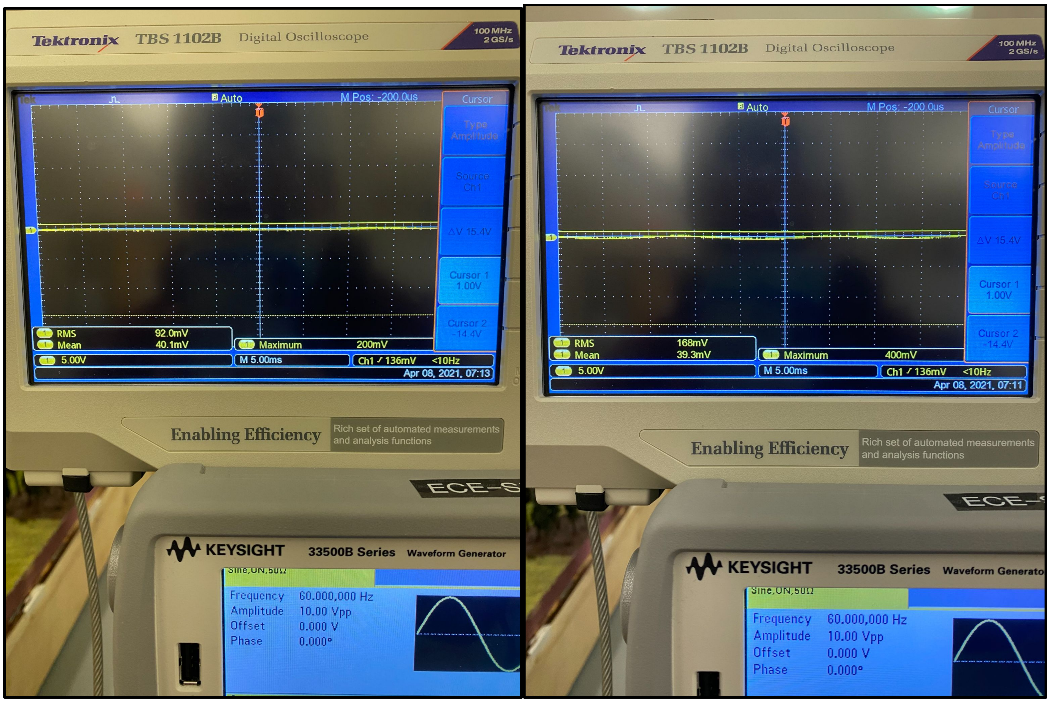
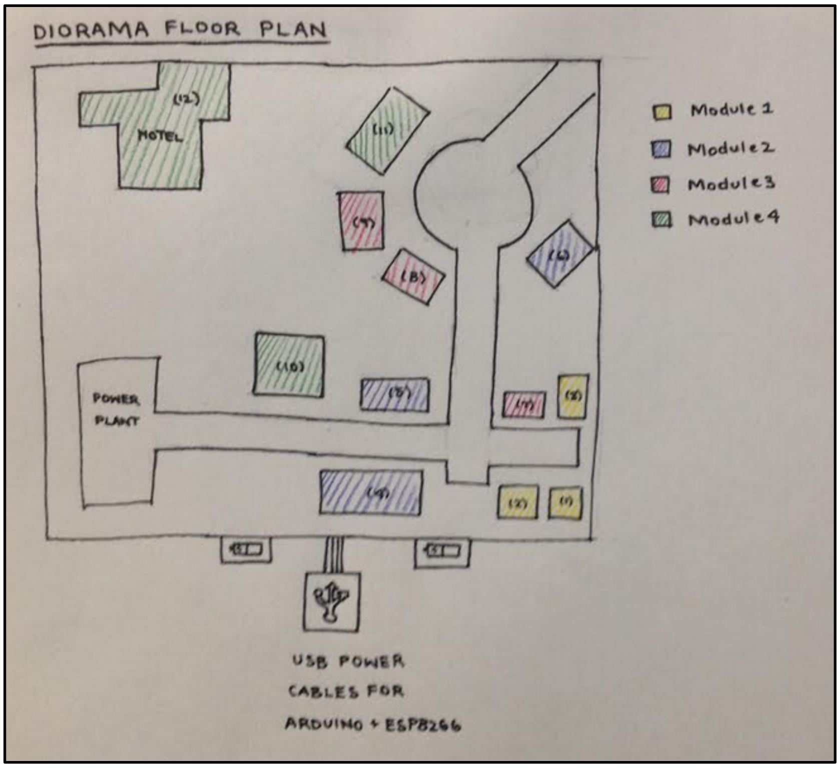


Figure.6 : Voltages at Oscilloscope Node (Relay Control Voltage - High)
Left : Electric Vehicle
Right: Electric Heating

Figure.7: Diorama Physical Schematic



As seen in the figure to the right, the diorama is split into 4 submodules with 3 buildings in each one. As electric heating adoption increases from 0% to 50%, houses will gradually have electric heating starting from Module 1. Once Module 1's houses are all electric then houses in Module 2 will have electric heating, and so on. Once 50% is reached all of Module 4 will have electric heating, as this module's larger houses makes up 50% of the electric heating load. Above 50% the pattern continues similarly to prior to 50% with the inclusion of all of Module 4 on electric heating.

Figure.8: Wiring of Arduino Control

As seen in the image to the left, the wiring for all 24 three terminal LEDs results in a significant amount of wiring. As such breadboards were used for simplicity. Note the wiring for all components were color coded for easier troubleshooting.